A multidisciplinary design optimization algorithm with distributed autonomous subsystems

7th World Congress on Structural and Multidisciplinary Optimization
COEX Seoul, 21 May - 25 May 2007, Korea

A Multidisciplinary Design Optimization Algorithm with Distributed Autonomous SubSystems

Akshay Gupte[1], Samy Missoum[2], Suvrajeet Sen[3] and Jitamitra Desai[1]

[1]Systems & Industrial Engineering, University of Arizona, Tucson, USA
akshay@email.arizona.edu, jdesai@email.arizona.edu
[2]Aerospace & Mechanical Engineering, University of Arizona, Tucson, USA
smissoum@ame.arizona.edu
[3]Industrial, Welding & Systems Engineering, Ohio State University, Columbus, USA
sen.22@osu.edu

1. Abstract
This work in progress presents a novel Multidisciplinary Design Optimization (MDO) algorithm that is tailored to provide maximum autonomy to the disciplines (subsystems). The link between system and subsystem levels is performed through shared variables (resources) whose nominal values are not imposed on the subsystems. As opposed to approaches such as Collaborative Optimization, the disciplines have the ability to not only find the optimal values of their own local design variables but also to inform the system level of optimal values of shared variables viewed from the subsystems. A distributed autonomous formulation of the fully integrated MDO problem is provided using a penalty decomposition method and a trust region approach. Disciplinary feasibility is achieved by the addition of artificial variables. Within each iteration loop, exactly one subsystem and the system are optimized. A first-order approximation of each discipline is used at the system level and the quality of this approximation is measured by an update scheme for the trust region. Computational results for this work in progress are provided for two test problems.
2. Keywords: multidisciplinary design optimization, distributed autonomous subsystems, penalty decomposition, trust region, artificial variables

3. Introduction

Over the course of last few decades, *Multidisciplinary Design Optimization* **(MDO)** has received a lot of attention from researchers in various fields and has been the subject of numerous publications, particularly from the applied design optimization community. The reasons for this acute interest are two-fold. First, many real-world engineering design applications require the interaction of several subsystems, thus making the development of MDO tools a necessity to perform design optimization. The second reason lies in the mathematical, algorithmic and computational aspects of MDO that have given rise to many interesting challenges pertaining to the sharing/coupling of information between subsystems.

Broadly stated, MDO deals with formulating and solving problems arising in complex organizational systems that typically comprise of the so-called *system level* (or *management level*) coupled together with several *subsystems* (or *disciplines*), along with their interactions. In a classical MDO framework, each subsystems represents a discipline that is concerned with one aspect in the design of the complete system (e.g. thermodynamics, structures, aerodynamics etc.). The challenge lies in the ability of each subsystem to optimally exploit the available system-level (global) resources while simultaneously satisfying system-level objectives and subsystem autonomy. A detail discussion of formulation methodologies and computational costs for coupled systems is provided in [1]. Due to the presence of the inter-disciplinary coupling, the optimal decisions made at the subsystem level might have conflicting objectives, thus increasing the challenge of developing an efficient approach. Then in a game theoretic perspective, each discipline plays the role of a player in the game. The interactions can then be modeled as Pareto (cooperative), Nash (noncoopeative) or Stackelberg (leader-follower) formulations depending on the level of cooperation and information sharing [2].

The disciplinary analysis phase is sometimes computationally intensive and time-consuming. Response Surface Methodologies (RSM) and other meta-modeling techniques have been of considerable help in solving issues related to high computational time, black-box, discontinuous and non-smooth functions [3]. A typical example is the trade-off that occurs in the case of optimizing car design simultaneously for NVH (noise, vibration and harshness) and crashworthiness [4]. A high cost of the

crash response is approximated by a meta-model whereas linear dynamic finite element analysis can be implemented to determine NVH parameters within an optimization loop.

There have been several attempts to develop modeling frameworks and algorithms for MDO. Almost all the MDO algorithms face issues of algorithmic convergence, feasibility, consistency of solution and attainment of a Karush-Kuhn-Tucker point. Of particular interest are the techniques such as *Collaborative Optimization* (CO) [5], *Bi-Level Integrated System Synthesis* (BLISS) [6], *Concurrent SubSpace Optimization* (CSSO) [7, 8], *Simultaneous Analysis and Design* (SAND) [1, 9] and *Multidisciplinary Design Optimization with Independent Subspaces*(MDOIS) [10]. All these methodologies vary primarily with respect to handling the interactions between the system & the subsystems and the degree of autonomy imparted to each subsystem. Also, most of these algorithms have been tested on real-world problems with varying degrees of success.

In this research effort, we propose a novel MDO algorithm that is based on *maximizing the subsystem autonomy* and is applicable to a general class of MDO problems. Although the necessity of imparting autonomy is clearly understood by the design community, approaches that are exclusively dedicated to this end are nonexistent. The central idea of our algorithm is to give each subsystem the autonomy to optimize its own local design variables as well as system (resource) variables, which are bounded within a *trust region*. Also, the system objective is composed of the subsystem objective functions. During the system-level optimization, since the system has decision-making control over only the system variables, the subsystem objectives are *linearized* with respect to their local variables. It is important to note that although each subsystem is linearized, the system objective can be a nonlinear function of these linearizations. Thus the system optimization independently revises the values of the global variables. A *trust region update* is used to guide the quality of linearizations at the system-level. Trust region models are frequently used in optimizing nonlinear functions whose first-order gradients are unavailable [11]. Using the revised updates the subsystems are re-optimized and the process is iterated until convergence.

The consistency constraints between the system resources and each subsystems' value for the system resources are relaxed and are penalized in the objective function of system and each subsystem using a quadratic *exterior penalty function*. This penalty parameter is updated using a suitable update scheme. To decouple the interdisciplinary coupling, each subsystem uses the values from the previous iteration of the local design variables of other subsystems. Multidisciplinary feasibility is attained at termination. It is important to note that this formulation is distributed in its nature and guarantees disciplinary autonomy. A discussion on distributed formulations is provided in [5, 12]. The advantages of the proposed MDO algorithm are that the subsystem optimization problems are completely autonomous and can be done concurrently within a distributed parallel computing framework.

The proposed MDO algorithm is distinct from existing methods in that each subsystem is allowed to recommend system-level variable values (disciplinary autonomy). Also, we use a trust region approach alongwith a penalty function formulation. The issue of disciplinary autonomy is addressed in CO, BLISS and CSSO. However CO involves imposing target values to the subsystem optimization problem while trying to minimize the deviation from this target, which essentially restricts the autonomy provided to the subsystem. On the contrary, our algorithm forms a trust region on the system variables and updates the trust region diameter at the system level. Both BLISS and CSSO use the Global Sensitivity Equations (GSE) in their algorithm and do not adopt a trust region framework. On a general basis, GSE may be impractical to implement considering the organizational challenges of the MDO framework. One subsystem may not have access to the sensitivities of another subsystem. However, as will be discussed in a future section, the knowledge of global sensitivities may be a crucial factor in the attainment of a Karush-Kuhn-Tucker (KKT) point. MDOIS [10] does not account for autonomy and uses GSE. [13, 14] provide some inspiring ideas on autonomy. However the narrow class of problems handled in [10, 13, 14] assumes that the subsystem problems involve local design variables and system resource variables but no variables from other subsystems, thus not accounting for the interdisciplinary coupling observed in many real-world problems. Including the interdisciplinary coupled variables as a part of the shared system variables may not be possible since one discipline may not have a decision-taking control over another discipline's design variables.

The remainder of this paper is organized as follows. In §4 we present the formulation of the MDO problem. In §5, the various elements and a step-by-step description of the algorithm are discussed in greater detail. Computational results for the test problems are presented in §6. §7 concludes the paper

with a summary.

4. Problem Formulation

The original system and subsystem formulations for the MDO problem are of the following form

$$\begin{array}{lll} & & \underline{\text{System} - \text{level}} \\ \min\limits_{s_0} & F_0 = & f_0(s_0, f_1(s_0, x_1 \dots x_N) \dots f_N(s_0, x_1 \dots x_N)) \\ \text{s.t.} & & s_0 \in S \cap \Omega \subseteq \Re^m \\ & & \Omega = \{s_0 : \ s_l \leq s_0 \leq s_u\} \\ \\ & & \underline{\text{Subsystem } i} \\ \min\limits_{x_i} & & f_i(s_0, x_1 \dots x_N) \\ \text{s.t.} & & g_i(s_0, x_1 \dots x_N) \ \ \leq 0 \\ & & h_i(s_0, x_1 \dots x_N) \ \ = 0 \\ & & x_i \in \Theta_i \subseteq \Re^{n_i} \\ \text{where} & & \Theta_i = \{x_i : \ x_{i_l} \leq x_i \leq x_{i_u}\} \end{array} \tag{1}$$

s_0 is the system-level variable, x_i is the local design variable of subsystem i, N is the number of subsystems, n_i is the number of local variables for subsystem i and $n = n_1 + \ldots + n_N$ is the total number of local design variables at subsystem level. The function $f_0 : \Re^{m+N} \rightarrow \Re$ is the system objective function while $f_i : \Re^{m+n} \rightarrow \Re$ is the objective function of subsystem i. $g_i : \Re^{m+n} \rightarrow \Re^{\mathbf{g_i}}$ and $h_i : \Re^{m+n} \rightarrow \Re^{\mathbf{h_i}}$ are the constraints on the problem of subsystem i and hence define its design search space. The functions f_0, f_i, g_i, h_i are assumed to be continuous and twice continuously differentiable. The fact that (f_i, g_i, h_i) are functions of $(x_1 \dots x_N)$ and not just x_i amply demonstrates the interdisciplinary coupling. The set S is some closed set that defines the design constraints on s_0 at the system-level. The sets Ω and Θ_i are hyper-rectangles in $\Re^m$ and $\Re^{n_i}$ respectively and hence are compact convex sets. Thus the set $S \cap \Omega$ is a compact set in $\Re^m$.

Given Eq.(1), the following steps are followed to relax problem (1) into a bilevel, autonomous, distributed formulation

1. Introduction of proxy system variables s_i at subsystem i and the corresponding consistency constraints $C_i \equiv s_0 - s_i = 0 \quad \forall i \in \{1 \dots N\}$
2. Relaxation of consistency constraints C_i by forming the exterior penalty function Φ_i where

$$\Phi_i(s_0, s_i) \ = \parallel (\sigma_i)^{1/2} \circ (s_0 - s_i) \parallel^2 \tag{2}$$

   where the symbol $\circ$ denotes the Hadamard product
3. Introduction of trust region constraints of the form $\parallel s_0 - s_i \parallel \ \leq \ \Delta_i$
4. Decomposition into a single system problem and N subsystem problems

The norms $\| \cdot \|$ used here are the l_2-norm. It must be noted that the solution to the relaxed problem is not equal to the solution to the original formulation (1). However the relaxation error can be driven to zero with an appropriate trust region and penalty update. The subsystem and system formulations are discussed next.

4.1 Subsystem optimization

Using Eq. (2), the relaxed consistency constraints for subsystem i at iteration k are given by

$$\Phi_i{}^k(s_i) = \parallel (\sigma_i{}^k)^{1/2} \circ (s_i - s_0{}^{k-1}) \parallel^2 \tag{3}$$

For the purpose of decoupling the interdisciplinary interaction, each subsystem i uses the values of $(\{x_j\}_{j=1}^N \ j \neq i)$ from the previous iteration. Thus at iteration k, $p_i{}^k = (\{x_j{}^{k-1}\}_{j=1}^N \ j \neq i)$ appears as a parameter in the optimization problem $\Pi_i{}^k$ of subsystem i. Also, it is observed that the presence of equality constraints h_i forces the variables (s_i, x_i) to strictly follow a certain manifold, which may cause

problem ${\Pi_i}^k$ to be infeasible for the given values of ${p_i}^k$. The inequality constraints g_i allow more search space for the optimization algorithm. Hence we introduce slack variables $\xi_i \in \Re_+^{\mathbf{g_i}}$ and unrestricted artificial variables $\zeta_i \in \Re^{\mathbf{h_i}}$ and put a penalty term for ζ_{i_j} in the objective function. The value of s_0 is fixed in problem Π_i. Also the knowledge of system design constraints defined by S in problem (1) is available only at the system-level and hence is not imposed on the subsystem. The formulation for problem Π_i at iteration k is given by

$$\begin{aligned} {\Pi_i}^k \equiv \min_{(s_i,x_i,\xi_i,\zeta_i)} \quad F_i = \quad & f_i(s_i,x_i,{p_i}^k) + \tfrac{1}{2}{\Phi_i}^k(s_i) + \tfrac{M_i}{2}{\zeta_i}^T\zeta_i \\ \text{s.t.} \quad & \| \, s_i - {s_0}^{k-1} \, \| \quad \leq {\Delta_i}^k \\ & g_i(s_i,x_i,{p_i}^k) + \xi_i \quad = 0 \\ & h_i(s_i,x_i,{p_i}^k) + \zeta_i \quad = 0 \\ & s_i \in \Omega \subseteq \Re^m \\ & x_i \in \Theta_i \subseteq \Re^{n_i} \\ & \xi_i \in \Re_+^{\mathbf{g_i}} \;\; \zeta_i \in \Re^{\mathbf{h_i}} \end{aligned} \tag{4}$$

where M_i is some large positive scalar.

4.2 System Optimization

In problem (1) it is observed that the system objective $f_0 = f_0(s_0, f_1 \dots f_N)$ is implicitly a function of the local design variables x_i. Realizing that the system will not have decision-making control over x_i in a distributed autonomous framework, we linearize the function f_i at the system level and represent it by ${f_i}^L$. Then the transformed system objective is represented as

$${f_0}^L = f_0(s_0, {f_1}^L \dots {f_N}^L) \tag{5}$$

Also, the artificial variables ζ_i are linearized to ${\zeta_i}^L$ in order to drive ζ_i to zero at the subsystem level. The linearization schemes are presented in §5.1. In case there exist any explicit constraints on the objective function at the subsystem level, denoted by set ψ_i, they are imposed on the linearized objective ${f_i}^L$ at the system-level. The formulation for problem Π_0 at iteration k is given by

$$\begin{aligned} {\Pi_0}^k \equiv \min_{s_0} \quad F_0 = \quad & {f_0}^{k^L} + \tfrac{1}{2}\sum_{i=1}^{N}{\Phi_i}^k(s_0) + \frac{M_i}{2}{\zeta_i}^{k^L\,T}{\zeta_i}^{k^L} \\ \text{s.t.} \quad & \| \, s_0 - {s_0}^{k-1} \, \| \quad \leq {\Delta_i}^k \\ & s_0 \in S \cap \Omega \subseteq \Re^m \\ & {f_i}^L \in \psi_i \end{aligned} \tag{6}$$

The exterior penalty function is given by

$$\Phi_i{}^k(s_0) = \| \, (\sigma_i{}^k)^{1/2} \circ (s_0 - s_i{}^k) \, \|^2 \tag{7}$$

at the system level.

5. MDO Algorithm

The proposed MDO algorithm performs one system optimization and one subsystem (say i) optimization during each iteration k. This signifies that during each iteration, the system tries to find its own descent direction in reaction to the changes in decision made by subsystem i. It gauges the improvements made by the subsystem through its linearized objective function ${f_i}^L$. The linearizations from other subsystems j are held fixed at their values from the previous iteration. Also, each subsystem i maintains its own trust region Δ_i and penalty σ_i which are updated so $\{s_i\} \rightarrow s_0$. §5.1-5.4 describe the various elements of the algorithm with a detail mathematical description of the algorithm provided in §5.5.

5.1 Subsystem Sensitivity and Linearizations

Subsystem i is linearized around the point (s_{ik}, x_{ik}) using Taylor's first-order expansion

$${f_i}^{k^L} = {f_i}^k + (\nabla_{s_i}{f_i}^k)^T(s_0 - {s_i}^k) + (\nabla_{x_i}{f_i}^k)^T\nabla_{s_i}{x_i}^k(s_0 - {s_i}^k) \tag{8}$$

The linearized system objective at iteration k is then given by : ${f_0}^{k^L} = f_0(s_0, {f_1}^{k^L} \dots {f_N}^{k^L})$ which is a function of only s_0. Similarly the linearization of artificial variables ζ_i is derived as follows

$$\begin{aligned} \zeta_{i_j} &= -h_{i_j}(s_i, x_i, p_i) \qquad\qquad \forall j \in \{1 \dots \mathbf{h_i}\} \\ \therefore\ {\zeta_{i_j}}^{k^L} &= -{h_{i_j}}^{k^L} \\ &= -[{h_{i_j}}^k + (\nabla_{s_i} {h_{i_j}}^k)^T (s_0 - {s_i}^k) \\ &\qquad +(\nabla_{x_i} {h_{i_j}}^k)^T \nabla_{s_i} {x_i}^k (s_0 - {s_i}^k)] \end{aligned} \tag{9}$$

The first-order gradients $(\nabla f_i, \nabla h_i)$ can be evaluated either using analytical expressions, finite-differences or by Response Surface Approximations. The sensitivity of local variables x_i with respect to system resources s_i $(\nabla_{s_i} x_i)$ is approximated using the scheme given below.

Consider the restricted Lagrangian $\bar{L}(s_i, x_i)$ for subsystem i at iteration k (it is called restricted since the Lagrange multipliers (λ_i, μ_i) are fixed at their optimal values $(\bar{\lambda_i}, \bar{\mu_i})$ obtained after solving problem (4)). Let $\bar{L}_Q^k$ denote its quadratic approximation in a δ-neighbourhood of $({s_i}^k, {x_i}^k)$. Then

$$\bar{L}_Q^k(s_i, x_i) = \bar{L}({s_i}^k, {x_i}^k) + \nabla \bar{L} \begin{bmatrix} s_i - {s_i}^k \\ x_i - {x_i}^k \end{bmatrix} + \frac{1}{2} \left[(s_i - {s_i}^k)\ (x_i - {x_i}^k) \right] \nabla^2 \bar{L} \begin{bmatrix} s_i - {s_i}^k \\ x_i - {x_i}^k \end{bmatrix} \tag{10}$$

Let $\Delta s_i = (s_i - {s_i}^k)$ and $\Delta x_i = (x_i - {x_i}^k)$. Since $({s_i}^k, {x_i}^k)$ is a KKT point of subsystem i at iteration k with multipliers $(\bar{\lambda_i}, \bar{\mu_i})$, in a δ-neighbourhood of $({s_i}^k, {x_i}^k)$ we have

$$\frac{\partial \bar{L}_Q^k}{\partial s_i} = 0 \ \text{ and } \ \frac{\partial \bar{L}_Q^k}{\partial x_i} = 0 \implies \nabla^2 \bar{L} \begin{bmatrix} \Delta s_i \\ \Delta x_i \end{bmatrix} = 0 \tag{11}$$

Denote $\nabla^2 \bar{L} = \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix}$ where the columns of $\begin{bmatrix} A_i \\ C_i \end{bmatrix}$ correspond to $(s_i - {s_i}^k)$ and those of $\begin{bmatrix} B_i \\ D_i \end{bmatrix}$ correspond to $(x_i - {x_i}^k)$. Diagonalize submatrix B_i into an identity matrix $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ using Gauss elimination. Let the submatrix A_i be transformed to ${A_i}^*$. Then we have

$${A_i}^*(\Delta s_i) + I(\Delta x_i) = 0 \implies \nabla_{s_i} x_i = -{A_i}^* \tag{12}$$

5.2 Trust Region update

Approximations play an important role in MDO by providing information to the system-level at a relatively low computational cost. [15] discusses some of the first-order approximation models used in engineering optimization. Trust region method is one such technique for management of approximation models and encompasses the idea of imposing move limits on variables by using a reduction/magnification scheme based on the quality of previous iterand [16].

In our MDO algorithm we manage a trust region model at the system level based on the approximations used in Eq. (5). A trust radius Δ_i is managed for each subsystem i. Let ${f_0}^{L^*}$ be the optimal linearized system objective and s_0^* be the optimal system variable. Then the relative reduction is

$$\rho = \frac{{f_0}^{k-1} - {f_0}^{k^*}}{{f_0}^{k-1} - {f_0}^{L^*}} \tag{13}$$

where ${f_0}^{k^*} = f_0(s_0^*, \{f_i(s_0^*, {x_i}^k, {p_i}^k)\})$ is the actual system objective value. If subsystem i has optimized its local problem in iteration k, then the following update scheme is used $(0 \le \eta_2 < \eta_1 \le 1)$

$$\begin{array}{lllll} \text{If} & \rho \ge \eta_1 \implies & {s_0}^k = s_0^*\,,\ {f_0}^k = {f_0}^{k^*} & {\Delta_i}^{k+1} = \lambda_1 {\Delta_i}^k & (\lambda_1 > 1) \\ \text{Elseif} & \rho \ge \eta_2 \implies & {s_0}^k = s_0^*\,,\ {f_0}^k = {f_0}^{k^*} & {\Delta_i}^{k+1} = {\Delta_i}^k & \\ \text{Else} & \implies & {s_0}^k = {s_0}^{k-1}\,,\ {f_0}^k = {f_0}^{k-1} & {\Delta_i}^{k+1} = \lambda_2 {\Delta_i}^k & (\lambda_2 < 1) \end{array} \tag{14}$$

5.3 Penalty parameter update

Exterior penalty methods are sometimes used for nonlinear programming problems with equality

constraints [16]. [14] uses the augmented Lagrangian algorithm to solve MDO problems that do not have an interaction between subsystems. In this MDO algorithm, we use the exterior penalty decomposition method. The same penalty parameter ${\sigma_i}^k$ is used in subsystem i and the system-level at iteration k. The following penalty update is analogous to the outer loop of the augmented Lagrangian algorithm.

$$\begin{aligned}
&\text{Define } viol({\Phi_i}^k) = \| {s_i}^k - {s_0}^k \|_\infty = \max_{m=1\ldots M} \{|{s_{i_m}}^k - {s_{0_m}}^k|\} \\
&\text{If } k \geq 2 \text{ then } \forall i = 1 \ldots N \\
&\quad \text{If } viol({\Phi_i}^k) \geq viol({\Phi_i}^{k-1}) \text{ then} \\
&\qquad \forall m = 1 \ldots M \ni |{s_{i_m}}^k - {s_{0_m}}^k| > viol({\Phi_i}^{k-1}) \\
&\qquad\quad \implies {\sigma_{i_m}}^{k+1} = 2{\sigma_{i_m}}^k
\end{aligned} \tag{15}$$

5.4 Convergence criteria

The MDO algorithm terminates when the system and all the subsystems agree on the value of the system resources i.e.

$$\| {s_0}^k - {s_i}^k \|^2 < \epsilon \ , \| {s_i}^k - {s_j}^k \|^2 < \epsilon \ \ \forall i, j \in \{1 \ldots N\} \tag{16}$$

where ϵ is some small positive scalar

5.5 Description of the algorithm

1. *Initialization Step* Let N be the number of subsystems. Set the iteration counter $k = 1, i = 1$ and initialize the values of $s_0, (x_1 \ldots x_N)$ and Δ_i
2. *Subsystem Optimization* For subsystem i, evaluate the parameter p_i by passing the values of ${x_j}^{k-1}$ from all other subsystems j. Pass the value of ${s_0}^{k-1}$ from the system level and solve the optimization problem ${\Pi_i}^k$ given by (4). The optimal values are $({f_i}^k, {s_i}^k, {x_i}^k, {\xi_i}^k, {\zeta_i}^k)$. Update the value of ${f_0}^{k-1}$ using the value ${f_i}^k$
3. *Subsystem Sensitivities* Evaluate the sensitivities $\nabla {f_i}^k$ and $\nabla_{s_i} {x_i}^k$ using Eq. (12)
4. *Subsystem Linearization* Linearize the subsystem i objective ${f_i}^{k^L}$ and artificial variables ${\zeta_i}^{k^L}$ based on Eq. (8) and Eq. (9). For all other subsystems j, set the linearizations ${f_j}^{k^L} = {f_j}^{k-1^L}$ and similarly for ${\zeta_j}^{k^L}$. If $k < N$, set ${f_j}^{k^L} = 0 \ \forall j > i$
5. *System Optimization* Pass the linearizations from the subsystem level to the system level and solve the optimization problem ${\Pi_0}^k$ given by (6)
6. *Trust Region update* Perform the trust region update on ${\Delta_i}^k$ using Eq. (14)
7. *Counter Increment* Set $k = k + 1$ and $i = i + 1$. If $i = N$ go to Step 8, else go to Step 9
8. *Check Termination* Check the convergence criteria in Eq. (16). If it is satisfied, STOP else go to Step 9
9. *Penalty update* Update the penalty parameters ${\sigma_i}^k$ for subsystem i as given in Eq. (15). Go to Step 2

6. Computational Results

The proposed MDO algorithm is validated using two test problems. The SQP algorithm available in MATLAB Optimization toolbox is used as a plugin to the object-oriented code developed in C++. Analytical derivatives are used for both the test problems.

6.1 Heart Dipole Problem

The Heart Dipole problem is a test problem in the NASA MDO test suite for evaluating the performance of MDO algorithms [17]. It has no pre-defined system and subsystems and can be stated as a set of eight nonlinear equations (feasibility problem) in eight variables $(x_1 \ldots x_8)$ given eight pieces

of measured data (parameters d). This problem arises from the experimental electrolytic determination of the resultant dipole moment in the heart.

$$
\begin{array}{ll}
f_1 & \equiv x_1 + x_2 - d_7 = 0 \\
f_2 & \equiv x_3 + x_4 - d_8 = 0 \\
f_3 & \equiv x_1x_5 + x_2x_6 - x_3x_7 - x_4x_8 - d_1 = 0 \\
f_4 & \equiv x_1x_7 + x_2x_8 + x_3x_5 + x_4x_6 - d_2 = 0 \\
f_5 & \equiv x_1({x_5}^2 - {x_7}^2) - 2x_3x_5x_7 + x_2({x_6}^2 - {x_8}^2) - 2x_4x_6x_8 - d_3 = 0 \\
f_6 & \equiv x_3({x_5}^2 - {x_7}^2) + 2x_1x_5x_7 + x_4({x_6}^2 - {x_8}^2) + 2x_2x_6x_8 - d_4 = 0 \\
f_7 & \equiv x_1x_5({x_5}^2 - 3{x_7}^2) + x_3x_7({x_7}^2 - 3{x_5}^2) + x_2x_6({x_6}^2 - 3{x_8}^2) + x_4x_8({x_8}^2 - 3{x_6}^2) - d_5 = 0 \\
f_8 & \equiv x_3x_5({x_5}^2 - 3{x_7}^2) - x_1x_7({x_7}^2 - 3{x_5}^2) + x_4x_6({x_6}^2 - 3{x_8}^2) + x_2x_8({x_8}^2 - 3{x_6}^2) - d_6 = 0
\end{array}
\tag{17}
$$

The given system of equations in Eq. (17) is decomposed into four subsystems - A, B, C, D, and one system as shown below

$$
\begin{array}{ccccc}
\textit{System} & \textit{Subsystem } A & \textit{Subsystem } B & \textit{Subsystem } C & \textit{Subsystem } D \\
s = \left[\begin{array}{cc} x_1 & x_3 \end{array} \right] & x_A = \left[\begin{array}{cc} x_5 & x_7 \end{array} \right] & x_B = \left[\begin{array}{cc} x_6 & x_8 \end{array} \right] & x_C = x_2 & x_D = x_4 \\
\min\ f_5 + f_6 + f_7 + f_8 & \min\ f_5 & \min\ f_6 & \min\ f_7 & \min\ f_8 \\
 & f_5 \geq 0 & f_6 \geq 0 & f_7 \geq 0 & f_8 \geq 0 \\
 & f_3 = 0 & f_4 = 0 & f_1 = 0 & f_2 = 0
\end{array}
\tag{18}
$$

The decomposition in Eq. (18) is relaxed into a distributed autonomous formulation using Eq. (4) and Eq. (6). Upper and lower bounds of 10 and -10 are imposed on the subsystem local variables (x_A, x_B, x_C, x_D). The high degree of nonlinearity and nonconvexity of the functions demonstrates the challenge in finding a optimal solution. The Heart Dipole problem is successfully solved for three instances with different initial value conditions and parameter values. We present computational results for three instances solved with the parameter values $d = 1$ and $d = 0$ in Table 1 using $\epsilon = 0.01, \lambda_1 = 1.2, \lambda_2 = 0.8, \eta_1 = 0.65, \eta_2 = 0.2$. Initial trust diameter values of 2.23 and 1.5 were used for the two instances. At termination, the trust diameters for the four subsystems were $(3.08, 3.69, 3.85, 1.37)$ and $(5.7, 1.69, 3.65, 4.56)$ in the two problem instances. All the equations / functions were equal to zero within tolerance at optimality. It must be noted that the initial value of $x = x^0$ is infeasible to the set of eight nonlinear equations.

Table 1: Numerical results for Heart Dipole problem

Problem	x^0 = Initial x	x^* = Optimal x
$d = 1$	$s_0 = \left[\begin{array}{c} 0 \\ 2 \end{array} \right]$ $x_A = \left[\begin{array}{c} 1 \\ -1 \end{array} \right]$ $x_B = \left[\begin{array}{c} 3 \\ 0 \end{array} \right]$ $x_C = 2$ $x_D = -1$	$s_0 = \left[\begin{array}{c} 0.0022 \\ 0.002 \end{array} \right]$ $x_A = \left[\begin{array}{c} 1.65 \\ -2.11 \end{array} \right]$ $x_B = \left[\begin{array}{c} 1.01 \\ 0 \end{array} \right]$ $x_C = 0.998$ $x_D = 0.997$
$d = 1$	$s_0 = \left[\begin{array}{c} 1 \\ 2 \end{array} \right]$ $x_A = \left[\begin{array}{c} 0 \\ 1 \end{array} \right]$ $x_B = \left[\begin{array}{c} 2 \\ 3 \end{array} \right]$ $x_C = 3.2$ $x_D = 1.3$	$s_0 = \left[\begin{array}{c} 0 \\ 0 \end{array} \right]$ $x_A = \left[\begin{array}{c} 3.542 \\ 1.38 \end{array} \right]$ $x_B = \left[\begin{array}{c} 1.005 \\ 0 \end{array} \right]$ $x_C = 1$ $x_D = 1$
$d = 0$	$s_0 = \left[\begin{array}{c} 1 \\ 2 \end{array} \right]$ $x_A = \left[\begin{array}{c} 0 \\ 1 \end{array} \right]$ $x_B = \left[\begin{array}{c} 2 \\ 3 \end{array} \right]$ $x_C = 3.2$ $x_D = 1.3$	$s_0 = \left[\begin{array}{c} 0.0185 \\ 0.0089 \end{array} \right]$ $x_A = \left[\begin{array}{c} 1.76 \\ -1.167 \end{array} \right]$ $x_B = \left[\begin{array}{c} 3.25 \\ 0 \end{array} \right]$ $x_C = -0.0015$ $x_D = 0.0053$

The convergence graphs for problem instance 1 are presented in Figure (1) - (3).

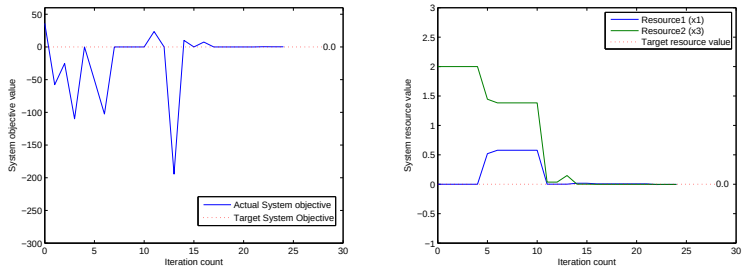

Figure 1: Convergence of system objective and system resources for Heart Dipole

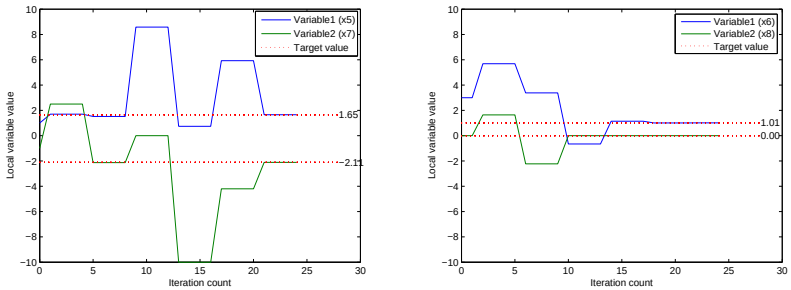

Figure 2: Convergence of local variables for subsystems A and B for Heart Dipole

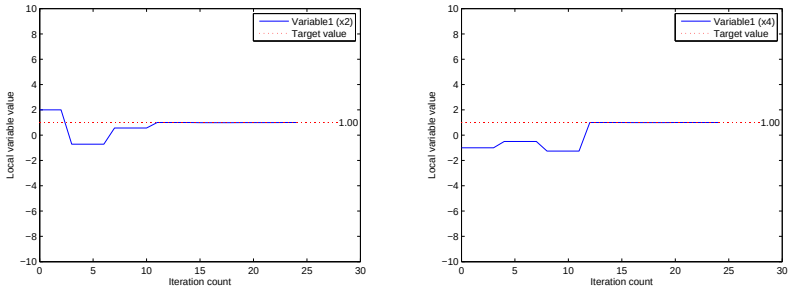

Figure 3: Convergence of local variables for subsystems C and D for Heart Dipole

6.2 Test Problem 2

Example 1 from [10] is used as the second test problem for our MDO algorithm. However the problem is modified in the sense that the original equality constraints $h_1 = 0$ and $h_2 = 0$ are replaced by inequality constraints $h_1 \leq 0$ and $h_2 \leq 0$. There are no predefined system and subsystems. The NLP

formulation is given below

$$
\begin{array}{lll}
\min\limits_{(x_1 \ldots x_4, y_1, y_4)} & F = & y_2 + y_3 \\
\text{s.t.} & h_1 = & \frac{x_1^2 + 2x_1x_2 + x_3x_4 - 3}{25} - y_1 - 4.2y_4 \leq 0 \\
& h_2 = & \frac{x_1x_4 + x_2x_3 + x_3x_4 - 10}{20} + 2y_1 - y_4 \leq 0 \\
& g_1 = & x_1x_2 + x_2^2 + y_4 - 3 \leq 0 \\
& g_2 = & x_4^2 - 4x_3x_4 - 2x_4 + y_1 + 7 \leq 0 \\
& g_3 = & x_3^2 + x_4 - x_3y_1 - 3 \leq 0 \\
& & -10 \leq x_1 \ldots x_4 \leq 10 \\
\text{where} & y_2 = & x_1^2 + 2x_1x_2^4 + 5x_2 + x_1 \exp y_4 + 12 \\
& y_3 = & x_4^2 + 2x_3{x_4}^2 - 8x_3 + x_4y_1^2
\end{array}
\tag{19}
$$

Eq. (19) is decomposed into one system and two subsystems - A, B as follows

$$
\begin{array}{ccc}
\textit{System} & \textit{Subsystem } A & \textit{Subsystem } B \\
s = \begin{bmatrix} y_1 & y_4 \end{bmatrix} & x_A = \begin{bmatrix} x_1 & x_2 \end{bmatrix} & x_B = \begin{bmatrix} x_3 & x_4 \end{bmatrix} \\
\min\ y_2 + y_3 & \min\ y_2 & \min\ y_3 \\
 & h_1 \leq 0 & h_2 \leq 0 \\
 & g_1 \leq 0 & g_2 \leq 0 \ \ g_3 \leq 0 \\
 & -10 \leq x_1, x_2 \leq 10 & -10 \leq x_3, x_4 \leq 10
\end{array}
\tag{20}
$$

The computational results for the above problem are presented in Table 2. The values of the problem parameters $(\epsilon, \lambda_1, \lambda_2, \eta_1, \eta_2)$ are the same as those in §6.1. An initial trust diameter of 5.5 is used. At termination, the trust diameters for the two subsystems were $(0.42, 0.8)$. The convergence graphs are presented in Figure (4) & (5).

Table 2: Numerical results for test problem 2

x^0 = Initial x	x^* = MDO optimal x	MDO optimal f_0	KKT optimal f
$s_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $x_A = \begin{bmatrix} -1 \\ -2 \end{bmatrix}$ $x_B = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$	$s_0 = \begin{bmatrix} 0.42 \\ 0.276 \end{bmatrix}$ $x_A = \begin{bmatrix} -0.609 \\ -1.374 \end{bmatrix}$ $x_B = \begin{bmatrix} 1.648 \\ 0.973 \end{bmatrix}$	-8.594	-9.32

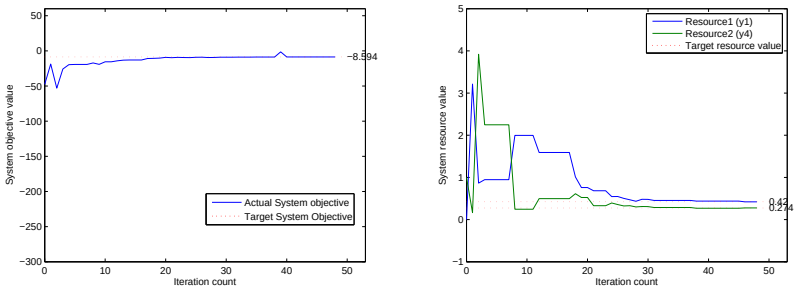

Figure 4: Convergence of system objective and system resources

The 7.7% optimality gap between the KKT local optimal value (-9.32) and the optimal value from our MDO algorithm (-8.594) is due to the fact that the MDO algorithm does not necessarily guarantee

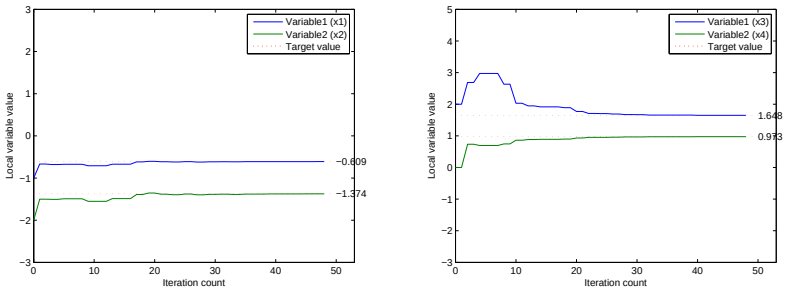

Figure 5: Convergence of local variables for subsystems A and B

a KKT point. Since a distributed autonomous framework does not allow the use of sensitivity of the Lagrangian of other subsystems (in other words GSE), there is no guarantee that the point obtained at termination will be a KKT equilibrium point.

7. Conclusion

An MDO algorithm with the objective of maximizing disciplinary autonomy has been developed. The algorithm is based on a trust-region approach that dictactes the availability of resources for each discipline. Within these boundaries, each discipline has the ability to not only make optimal decisions on its own local variables but also on the system variables. Consistency between disciplinary decisions is enforced through a penalty approach. The algorithm has been tested on two examples from the litterature.

Because the algorithm does not guarantee convergence to a solution satisfying the first order optimality conditions, the next step of this research consists of developing a measure allowing to detect how far the solution is from optimality. In addition, larger scale problems will be used to validate the approach.

8. References

[**1**] R.J. Balling and J. Sobieszczanski-Sobieski, Optimization of Coupled Systems: A Critical Overview of Approaches, *AIAA Journal*, 1996, 34(1), 6-17.

[**2**] K. Lewis and F. Mistree, Modeling Interactions in Multidisciplinary Design: A Game Theoretic Approach, *AIAA Journal*, 1997, 35(8), 1387-1392.

[**3**] Kodiyalam S and Sobieszczanski-Sobieski J. Bi-Level Intergrated System Synthesis with Response Surfaces. *Proc. of the 40th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 1999, St.Louis, MO.

[**4**] S. Kodiyalam and J. Sobieszczanski-Sobieski, Multidisciplinary design optimization - some formal methods, framework requirements, and applications to vehicle design, *International Journal of Vehicle Design*, 2001, 25(1-2), 3-22.

[**5**] N.M. Alexandrov and R.M. Lewis, Analytical and Computational Aspects of Collaborative Optimization for Multidisciplinary Design, *AIAA Journal*, 2002, 40(2), 301-309.

[**6**] J. Sobieszczanski-Sobieski, J.S. Agte and R.R. Sandusky Jr., Bilevel Integrated System Synthesis, *AIAA Journal*, 2000, 38(1), 164-172.

[**7**] B.A. Wujek, J.E. Renaud, S.M. Batill and J.B. Brockman, Concurrent Subspace Optimization Using Design Variable Sharing in a Distributed Computing Environment, *Concurrent Engineering*, 1996, 4(4), 361-377.

[**8**] Tappeta R, Nagendra S, Renaud J E and Badrinath K. Concurrent Sub-Space Optimization (CSSO) MDO Algorithms in iSIGHT. *Technical Report 97CRD187, GE Research & Development Center*, 1998.

[**9**] R.T. Haftka, Simultaneous analysis and design, *AIAA Journal*, 1985, 23(7), 1099-1103.

[**10**] M.K. Shin and G.J. Park, Multidisciplinary design optimization based on independent subspaces, *International Journal for Numerical Methods in Engineering*, 2005, 64, 599-617.

[**11**] Conn A R, Scheinberg K and Toint P L. A derivative-free optimization algorithm in practice. *Proc. of the 7th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998, St.Louis, MO, 129-139.

[**12**] Alexandrov N M and Lewis R M. Analytical and Computational Properties of Distributed Approaches to MDO. *Proc. of the 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2000, Long Beach, CA.

[**13**] R.T. Haftka and L.T. Watson, Multidisciplinary Design Optimization with Quasiseparable Subsystems, *Optimization and Engineering*, 2005, 6, 9-20.

[**14**] Tosserams S, Etman LFP and Rooda J E. An augmented Lagrangian decomposition method for quasi-separable problems in MDO. *Proc. of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, 2006, Portsmouth, Virginia.

[**15**] N.M. Alexandrov and R.M. Lewis, An Overview of First-Order Model Management for Engineering Optimization, *Optimization and Engineering*, 2001, 2(4), 413-430.

[**16**] Fletcher R. Practical Methods of Optimization. *2nd ed. Wiley Publications*, 2000.

[**17**] Padula S L, Alexandrov N M and Green L L. MDO test suite at NASA Langley Research Center. *Proc. of the 6th AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1996, Bellevue, WA.